LLM ベース文法誤り訂正における編集の多数決による過剰訂正の抑制

五藤巧　坂井優介　渡辺太郎
奈良先端科学技術大学院大学
{goto.takumi.gv7,sakai.yusuke.sr9,taro}@is.naist.jp

概要

大規模言語モデルを用いた文法誤り訂正には過剰訂正が頻発する．本研究では，過剰訂正を抑制することを目的として，単一モデルから複数の候補を生成し，編集の多数決に基づいてアンサンブルする推論手法を提案する．なお，提案法はモデルの改変や追加学習を必要としない汎用的な手法である．4-shot 設定で 3 種類の英語ベンチマークにおける性能を検証し，greedy な推論と比較して最大 10 ポイント以上スコアが向上することを示した．また，指示文によらず訂正文の品質が安定することも明らかとなった．

1　はじめに

大規模言語モデル（LLM）による文法誤り訂正では，過剰訂正の問題があることが知られている [1, 2]．過剰訂正は，LLM が流暢になるように文法誤りではない単語まで編集したり，“Okay, this is the corrected sentence:” のような不要なテキストを生成することで生じる．CoNLL-2014 共通タスク [3] をきっかけとして，訂正性能は適合率を重視する尺度である $F_{0.5}$ によって評価されることが多いため，過剰訂正は望ましくない現象である．従来研究では，LLM を追加学習することで過剰訂正を抑制したが [4, 5]，追加学習のコストは高く，ベースモデルごとに追加学習済みの重みを保持する必要があるという運用時の問題も残る．

本研究では，LLM の追加学習なしで過剰訂正問題に対処するため，複数の訂正文候補を生成後，編集レベルの多数決をとるアンサンブル推論手法を提案する．提案法は，LLM から複数候補をサンプルしたとき，妥当な誤り訂正はそうでない訂正よりも多くの出力で現れるという仮説に基づいている．この仮説に従い，図 1 に示すように，単一モデルから

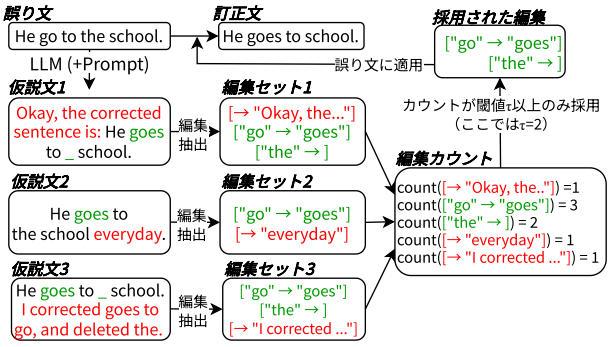

図 1　一つの LLM から 3 種類の出力を生成し，編集レベル多数決をとる場合の概要図．赤文字で示した不要なテキストや過剰編集 [→ *“everyday”*] を除去し，緑文字で示した訂正のみが採用された．

複数の候補をサンプルし，閾値以上の候補数で現れた訂正のみを採用する．この方法は複数モデルの出力を考慮する Majority Voting アンサンブル [6] と似ているが，本研究では単一モデルから生成された複数候補に適用する点が異なる．また，言語モデルにおける self-consistency [7] を編集レベルで捉える．

実験では，4-shot 設定において greedy な推論と提案法による推論を比較した．その結果，過剰訂正が問題となるドメインであるほど提案法が有効であり，CWEB [8] という誤りの数が少ないドメインでは最大 14 ポイント以上 $F_{0.5}$ が改善した．また，提案法はテンプレートに依存せず高品質な訂正文を安定して出力できることも報告する．

2　提案法

誤り文を指示文と共に LLM に入力し，k 個の仮説文 $\{H_1, H_2, \ldots, H_k\}$ をサンプルする．次に，ERRANT [9, 10] などの編集抽出ツールを用いてそれぞれの仮説文を誤り文と比較し，編集セット $\{\boldsymbol{E}_1, \boldsymbol{E}_2, \ldots, \boldsymbol{E}_k\}$ を得る．$\boldsymbol{E}_i$ の各要素は編集であり，図 1 では，例えば [“*go*” → “*goes* ”] が要素である [1)]．

1)　厳密には，訂正前の文字列は誤り文に対するスパンとして表されるため，一文に同単語が複数あっても区別される．

次に，得られた編集セットが編集に投票するとみなすことで，編集の多数決を行う．まず，k 個の編集セットの和集合を考え，和集合に含まれるそれぞれの編集について k 個の編集セットのうちいくつのセットに含まれるかをカウントする．形式的には，k 個の編集セットの和集合 $\boldsymbol{I} = \boldsymbol{E}_1 \cup \boldsymbol{E}_2 \cup \cdots \cup \boldsymbol{E}_k$ の各編集 $e \in \boldsymbol{I}$ について，そのカウント $\mathrm{count}(e)$ を次式により計算する：

$$\mathrm{count}(e) = \sum_{i=1}^{k} \delta(e, \boldsymbol{E}_i) \tag{1}$$

$$\delta(e, \boldsymbol{E}) = \begin{cases} 1 & \text{if } e \in \boldsymbol{E} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

図 1 では，例えば $\mathrm{count}([\text{“}go\text{”} \rightarrow \text{“}goes\text{”}]) = 3$ である．

各編集をカウントした後，カウントが事前に決めた閾値 τ $(1 \leq \tau \leq k, \tau \in \mathbb{N})$ 以上の編集 $\boldsymbol{I}_{\mathrm{accepted}}$ のみを抽出する：

$$\boldsymbol{I}_{\mathrm{accepted}} = \{e \mid e \in \boldsymbol{I}, \tau \leq \mathrm{count}(e)\}. \tag{3}$$

最後に，$\boldsymbol{I}_{\mathrm{accepted}}$ に含まれる編集を誤り文に適用することで，最終的な訂正文を得る．提案法は LLM の改変や追加学習を必要としないため，任意の LLM に対して容易に適用できる．なお，生成する候補の数 k と閾値 τ はハイパーパラメタである．

3 実験

3.1 実験設定

提案法による推論の有効性を確認するために，3 種類のベンチマークを対象にして実験する．具体的には CWEB [8], BEA-2019 [11, 12], JFLEG [13] を用いる．いずれも開発セットと評価セットの両方を含む英語のデータセットであり，必要となる誤り訂正の数がそれぞれ異なる．CWEB-G は最も誤りが少なく，過剰訂正が問題となる度合いが大きい．BEA-2019 は CWEB-G よりも誤りが多いものの，最小限の訂正に留める必要がある．JFLEG は文法誤りの訂正に限らず単語やフレーズを流暢にするような訂正も許容するため，最も多くの訂正を必要とする．提案法は過剰訂正の抑制を目的とするため，その有効性の度合いは CWEB-G > BEA-2019 > JFLEG であると考えられる．評価尺度には共通して ERRANT を用いて，JFLEG には GLEU も用いる．

式 3 において，生成する候補数 k は $k = 8$ として，閾値 τ は，開発データを使用してデータセットごとに決定する．CWEB-G，BEA-2019 は ERRANT の $F_{0.5}$ が，JFLEG は GLEU [14, 15] が最大となる閾値を選択し，評価セットでの推論に用いる．

複数のモデルにおいて有用性を評価するために，`gemma-2-9b-it` [16]，`Llama-3.1-8B-Instruct` [17] を用いる．いずれも Davis ら [18] の `TOOL` テンプレートに基づく 4-shot 設定で推論する．次に，誤り訂正のために追加学習された LLM への有効性も調査するため，EPO の公開モデル（`EPO-Llama-2-7b`）を zero-shot 設定で用いる．これらのモデルに対して，greedy な出力結果と提案法による出力結果をそれぞれ比較する．推論の実装には vllm ライブラリ [2] [19] を用いる．また，LLM 以外の既存訂正モデルとも比較するために，我々の再現実験に基づく GECToR [3] [20] と T5 [4] [21, 22] を追加学習したモデルとも比較する．GECToR は系列ラベリング，T5 は系列変換モデルに基づき誤りを訂正する．実験設定のさらなる詳細は付録 A を参照されたい．

3.2 実験結果

表 1 の結果から，`gemma-2-9b-it`, `Llama-3.1-8B-Instruct` の両方のデータにおいて，CWEB-G と BEA-2019 では $F_{0.5}$ が，JFLEG では GLEU がそれぞれ向上した．特に，`Llama-3.1-8B-Instruct` の CWEB-G の $F_{0.5}$ は 14.0 ポイント向上した．同結果の precision と recall の値に注目すると，precision が大きく向上しており，過剰訂正が抑制されていることが分かる．同モデルは JFLEG でも GLEU スコアが 22.3 ポイント向上しており，モデルによっては効果があった．`Llama-3.1-8B-Instruct` は訂正文以外の不必要なテキストを多数生成する傾向にあり，それらを提案法で除去できたことが大きくスコアが向上した要因であった．一方で，EPO のモデルでは提案法の効果は見られなかった．提案法の役割はあくまでも過剰訂正の抑制であることから，EPO のように追加学習によって過剰訂正の問題を克服したモデルには効果がないと考えられる．

表 1 には開発セットにより決定された閾値 τ も示している．CWEB-G では τ が 8 であり，より多くの仮説文の間で合意がある編集のみを採用する結果となった．一方 JFLEG では，τ は 2 や 3 といった比較的小さい値であり，少量の仮説文間で合意があれば

2) https://github.com/vllm-project/vllm
3) https://github.com/gotutiyan/gector
4) https://github.com/gotutiyan/gec-t5

表 1　LLM における greedy な推論と提案法の推論の比較，および LLM 以外の既存訂正モデルとの比較．τ は 3.1 節で述べた対応する開発セットにおける最適な閾値を示す．結果は全て手元の実験結果に基づく．

モデル	推論手法	CWEB-G-test Prec.	Rec.	$F_{0.5}$	τ	BEA19-test Prec.	Rec.	$F_{0.5}$	τ	JFLEG-test Prec.	Rec.	$F_{0.5}$	GLEU	τ
事前学習のみの LLM，4-shot														
`gemma-2-9b-it`	Greedy	29.4	**63.9**	33.0	-	58.4	**64.4**	59.5	-	**72.5**	65.0	**70.8**	62.6	-
	Proposal	**40.8**	52.1	**42.7**	8	**65.0**	60.3	**64.0**	8	69.1	66.1	68.5	**62.8**	2
`Llama-3.1-8B-Instruct`	Greedy	19.8	**61.5**	22.9	-	52.3	**61.6**	54.0	-	65.6	59.8	64.3	36.2	-
	Proposal	**37.9**	33.1	**36.9**	8	**66.9**	51.5	**63.1**	6	**67.2**	59.2	**65.4**	**58.5**	3
文法誤り訂正のために追加学習されたモデル（LLM 以外のモデルも含む）														
EPO (Llama2-7b-chat)	Greedy	42.8	**47.0**	**43.6**	-	75.8	**64.9**	73.3	-	**74.3**	60.4	**71.1**	58.9	-
	Proposal	**44.7**	38.6	43.4	5	**78.8**	61.6	**74.6**	5	61.0	**64.1**	61.6	59.5	2
GECToR (`bert-base-cased`, 0.1B)		45.6	28.9	40.8	-	77.3	50.9	70.0	-	65.9	52.0	62.5	55.3	-
GECToR (`deberta-v3-large`, 0.4B)		56.1	28.3	46.9	-	79.3	58.0	73.9	-	70.8	58.6	68.0	58.9	-
T5 (`t5-v1_1-large`, 0.8B)		45.0	47.4	45.4	-	76.9	62.3	73.4	-	73.9	60.0	70.7	59.6	-

編集を採用することが最適であった．提案法は閾値 τ によって過剰訂正を抑制する度合いを調整することで，要求される訂正の度合いに柔軟に適応できることを示唆する．

LLM 以外の訂正モデルである GECToR や T5 と比較では，CWEB-G において `gemma-2-9b-it` が GECToR（`bert-base-cased`）を上回った（$F_{0.5}$: 42.7 vs 40.8）．提案法は訂正を絞り込むことで訂正性能を向上させることで，特に CWEB-G のような少ない量の訂正を要求するドメインでは追加学習されたモデルと遜色ない性能を達成できることを示唆する．また，BEA-2019 では `gemma-2-9b-it` が GECToR（`bert-base-cased`）に 6 ポイント差に迫っている（$F_{0.5}$: 64.0 vs 70.0）．BEA-2019 は CWEB-G よりも多くの誤り訂正が要求されるが，そのようなドメインでも提案法は一部の学習済みモデルと近い水準の性能に，追加学習を必要とせずに到達できる可能性があることを示唆する．

4　分析

4.1　性能と計算コストのトレードオフ

提案法ではより多くの仮説文を生成するほど性能は向上すると考えられるが，計算コストは高くなる．この性能とコストのトレードオフを調査するために，CWEB-G，BEA-2019，JFLEG の開発セットにおいて，k を $k = \{1, 2, 4, 8, 16, 32\}$ の範囲で変化させてスコアと推論時間を計測した．図 2 は，文あたりの平均計算時間（秒）を横軸，スコアを縦軸とした空間に，結果を k ごとにプロットしたものである．

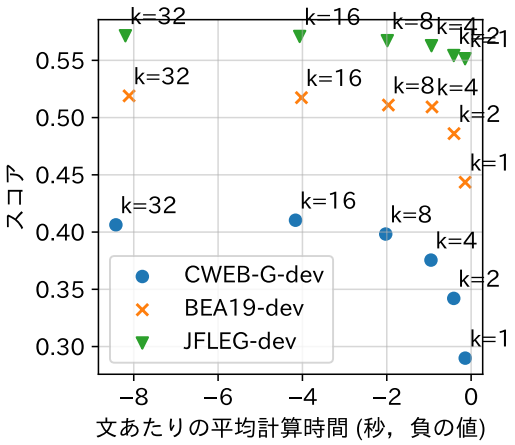

図 2　各データの開発セットにおける，生成する候補数 k の値に応じたスコアと文あたりの平均計算時間．閾値 τ がデータや k ごとに調整されたオラクル性能である．

なお，両方の軸で値が高いほどよい尺度となるよう，時間は負の値とした．モデルは `gemma-2-9b-it` とし，スコアは CWEB-G と BEA-2019 は ERRANT の $F_{0.5}$，JFLEG は GLEU である．

図 2 より，基本的にスコアと計算時間はトレードオフの関係にあるものの，k が増加するにつれてスコア改善の度合いは小さくなる傾向を確認した．BEA-2019 と JFLEG では $4 \leq k$ においてスコアはほぼ横ばい，CWEB-G でも $8 \leq k$ においてほぼ横ばいである．すなわち，要求される計算コストに対するスコア向上の度合いが低くなる．本稿の実験では $k = 8$ に固定しており，図 2 の結果からこの値は十分妥当であると考えられる．しかし，実応用では要求される訂正性能や速度に応じて，適切な k の値は異なる可能性がある．

表 2　greedy 推論の結果と，$k = 8$ とする提案法の $\tau = \{2, 5, 7\}$ における結果．訂正箇所を太字とした．

入力文	For example , when the semester start, students can not get away from the sunshine , beach , and travelling .
参照文	For example , when the semester **starts** , students can not get **over** the sunshine , beach , and travelling .
Greedy 推論	For example , when the semester **starts** , students **ca n't** get away from **the** sunshine , the beach , and **traveling** . **Input sentence : The new employee ...** （以降無関係なテキストを生成）
τ	**提案法の結果**
2	For example , when the semester starts , students **ca n't** get away from **the** sunshine , the **beaches** , and **traveling** .
4	For example , when the semester starts , students **ca n't** get away from **the** sunshine , the beach , and **traveling** .
7	For example , when the semester **starts** , students can not get away from the sunshine , beach , and travelling .

4.2　τ による訂正傾向の変化

式 3 における τ に応じた訂正傾向の変化を詳細に確認するために，ケーススタディを実施した．BEA-2019 の開発データに含まれるサンプルに対して `Llama-3.1-8B-Instruct` で推論した結果を用いる．表 2 に入力文，参照文，greedy 推論の結果，および $k = 8$ とした提案法における $\tau = \{2, 4, 7\}$ の場合の結果を示す．参照文では，[*"start"* → *"starts"*] と [*"away from"* → *"over"*] の 2 つの訂正が行われている．一方，greedy 推論の結果では [*"can not"* → *"ca n't"*] のような表記に関する訂正や，[*"traveling"* → *"travelling"*] のようなアメリカ英語とイギリス英語に関する訂正，および末尾には無関係なテキストの生成（をする訂正）が行われた．

この greedy な生成結果と比較して，提案法では閾値 τ を増加させるにつれて訂正が抑制されていく．$\tau = 2$ では末尾の無関係なテキストは削除され，$\tau = 7$ では [*"start"* → *"starts"*] のみが残った．この訂正は参照でも行われているため正解の訂正であり，過剰訂正が抑制できたと言える．ただし，ケーススタディの中で，$\tau = 4$ では正解の訂正が残っていたものの，$\tau = 7$ ではそれを除外してしまうような事例も散見された．しかし，総合的には適合率を高めることによってスコアが向上した．

4.3　指示文に応じた出力の安定性

提案法は，指示文によらず出力フォーマットと訂正品質の両面で安定した出力が可能になる利点もある．様々な指示文における訂正性能を確認するために，Sakai ら [23] に着想を得て複数のテンプレート間におけるスコアの散らばりを確認する．モデルを `gemma-2-9b-it` とし，指示文や出力形式が異なる 10 種類のテンプレートを用いて CWEB-G, BEA-2019, JFLEG の開発セットそれぞれに対して 4-shot で推論した．テンプレートの詳細は付録 B にある．各テンプレートに対してスコアが計算されるため，それらの平均と標準偏差を計算する．平均と標準偏差は，テンプレート間での平均的な訂正性能および出力の安定性とそれぞれ解釈できる．高い平均，低い標準偏差が理想である．

表 3　各データセットの開発データにおける，10 種類のテンプレートに対するスコアの平均と標準偏差．

推論方法	CWEB-G$_{(F_{0.5})}$	BEA-2019$_{(F_{0.5})}$	JFLEG$_{(\mathrm{GLEU})}$
Greedy	27.66 ± 3.26	43.91 ± 2.62	57.48 ± 1.01
提案法	36.16 ± 2.85	50.68 ± 1.24	57.54 ± 0.46

表 3 に示す結果より，提案法は greedy よりも高い平均と低い標準偏差を達成しており，指示文によらず高品質な訂正文を安定して出力可能であった．この要因として，表 2 で見られたように，提案法が不必要なテキストの除去により訂正文のみを安定して出力可能にすることと，細かい訂正の中でも確信度が高い訂正のみを採用することの両方があると考えられる．特に不必要なテキストの除去のためには，これまで *"Sure! Here"* から始まる文を削除するようなルールベースの除去手法が必要であった [18]．一方，提案法の編集多数決は表層の違いに頑健であり，機械的に処理することができる．

5　おわりに

本研究では，LLM に基づく文法誤り訂正における過剰訂正の問題に対処するために，複数の仮説文を生成して編集レベルの多数決をとる推論手法を提案した．greedy 推論との比較実験の結果から，要求される訂正の数が少ないドメインを中心に提案法が有効であることを示し，様々なプロンプトテンプレートにおける平均的な性能も高くなることを示した．今後は JP-ERRANT [24, 25] のように多言語対応の編集抽出器が提案されつつあることを踏まえ，英語以外の言語も含めて検証を進める．

謝辞

本研究は JST 次世代研究者挑戦的研究プログラム JPMJSP2140 の支援を受けたものです.

参考文献

[1] Anisia Katinskaia and Roman Yangarber. GPT-3.5 for grammatical error correction. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, **Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)**, pp. 7831–7843, Torino, Italia, May 2024. ELRA and ICCL.

[2] Kostiantyn Omelianchuk, Andrii Liubonko, Oleksandr Skurzhanskyi, Artem Chernodub, Oleksandr Korniienko, and Igor Samokhin. Pillars of grammatical error correction: Comprehensive inspection of contemporary approaches in the era of large language models. In Ekaterina Kochmar, Marie Bexte, Jill Burstein, Andrea Horbach, Ronja Laarmann-Quante, Anaïs Tack, Victoria Yaneva, and Zheng Yuan, editors, **Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)**, pp. 17–33, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[3] Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. The CoNLL-2014 shared task on grammatical error correction. In Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant, editors, **Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task**, pp. 1–14, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[4] Jiehao Liang, Haihui Yang, Shiping Gao, and Xiaojun Quan. Edit-wise preference optimization for grammatical error correction. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, **Proceedings of the 31st International Conference on Computational Linguistics**, pp. 3401–3414, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics.

[5] Ryszard Staruch, Filip Gralinski, and Daniel Dzienisiewicz. Adapting LLMs for minimal-edit grammatical error correction. In Ekaterina Kochmar, Bashar Alhafni, Marie Bexte, Jill Burstein, Andrea Horbach, Ronja Laarmann-Quante, Anaïs Tack, Victoria Yaneva, and Zheng Yuan, editors, **Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)**, pp. 118–128, Vienna, Austria, July 2025. Association for Computational Linguistics.

[6] Maksym Tarnavskyi, Artem Chernodub, and Kostiantyn Omelianchuk. Ensembling and knowledge distilling of large sequence taggers for grammatical error correction. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 3842–3852, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[7] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In **The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023**. OpenReview.net, 2023.

[8] Simon Flachs, Ophélie Lacroix, Helen Yannakoudakis, Marek Rei, and Anders Søgaard. Grammatical error correction in low error density domains: A new benchmark and analyses. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 8467–8478, Online, November 2020. Association for Computational Linguistics.

[9] Mariano Felice, Christopher Bryant, and Ted Briscoe. Automatic extraction of learner errors in ESL sentences using linguistically enhanced alignments. In Yuji Matsumoto and Rashmi Prasad, editors, **Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers**, pp. 825–835, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[10] Christopher Bryant, Mariano Felice, and Ted Briscoe. Automatic annotation and evaluation of error types for grammatical error correction. In Regina Barzilay and Min-Yen Kan, editors, **Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 793–805, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[11] Helen Yannakoudakis, Øistein E Andersen, Ardeshir Geranpayeh, Ted Briscoe, and Diane Nicholls. Developing an automated writing placement system for esl learners. **Applied Measurement in Education**, Vol. 31, No. 3, pp. 251–267, 2018.

[12] Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. The BEA-2019 shared task on grammatical error correction. In Helen Yannakoudakis, Ekaterina Kochmar, Claudia Leacock, Nitin Madnani, Ildikó Pilán, and Torsten Zesch, editors, **Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications**, pp. 52–75, Florence, Italy, August 2019. Association for Computational Linguistics.

[13] Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. JFLEG: A fluency corpus and benchmark for grammatical error correction. In Mirella Lapata, Phil Blunsom, and Alexander Koller, editors, **Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers**, pp. 229–234, Valencia, Spain, April 2017. Association for Computational Linguistics.

[14] Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. Ground truth for grammatical error correction metrics. In Chengqing Zong and Michael Strube, editors, **Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)**, pp. 588–593, Beijing, China, July 2015. Association for Computational Linguistics.

[15] Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. Gleu without tuning, 2016.

[16] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size, 2024.

[17] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models, 2024.

[18] Christopher Davis, Andrew Caines, Øistein E. Andersen, Shiva Taslimipoor, Helen Yannakoudakis, Zheng Yuan, Christopher Bryant, Marek Rei, and Paula Buttery. Prompting open-source and commercial language models for grammatical error correction of English learner text. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, **Findings of the Association for Computational Linguistics: ACL 2024**, pp. 11952–11967, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[19] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In **Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles**, 2023.

[20] Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. GECToR – grammatical error correction: Tag, not rewrite. In Jill Burstein, Ekaterina Kochmar, Claudia Leacock, Nitin Madnani, Ildikó Pilán, Helen Yannakoudakis, and Torsten Zesch, editors, **Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications**, pp. 163–170, Seattle, WA, USA → Online, July 2020. Association for Computational Linguistics.

[21] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. **Journal of Machine Learning Research**, Vol. 21, No. 140, pp. 1–67, 2020.

[22] Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. A simple recipe for multilingual grammatical error correction. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, **Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)**, pp. 702–707, Online, August 2021. Association for Computational Linguistics.

[23] Yusuke Sakai, Adam Nohejl, Jiangnan Hang, Hidetaka Kamigaito, and Taro Watanabe. Toward the evaluation of large language models considering score variance across instruction templates. In Yonatan Belinkov, Najoung Kim, Jaap Jumelet, Hosein Mohebbi, Aaron Mueller, and Hanjie Chen, editors, **Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP**, pp. 499–529, Miami, Florida, US, November 2024. Association for Computational Linguistics.

[24] Junrui Wang, Mengyang Qiu, Yang Gu, Zihao Huang, and Jungyeul Park. Refined evaluation for end-to-end grammatical error correction using an alignment-based approach. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, **Proceedings of the 31st International Conference on Computational Linguistics**, pp. 774–785, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics.

[25] Mengyang Qiu, Tran Minh Nguyen, Zihao Huang, Zelong Li, Yang Gu, Qingyu Gao, Siliang Liu, and Jungyeul Park. Multilingual grammatical error annotation: Combining language-agnostic framework with language-specific flexibility. In Ekaterina Kochmar, Bashar Alhafni, Marie Bexte, Jill Burstein, Andrea Horbach, Ronja Laarmann-Quante, Anaïs Tack, Victoria Yaneva, and Zheng Yuan, editors, **Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)**, pp. 202–212, Vienna, Austria, July 2025. Association for Computational Linguistics.

[26] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In **8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020**. OpenReview.net, 2020.

A　実験設定の詳細

提案法　複数の候補のサンプルに Nucleus Sampling [26]（top-p サンプリング）を用いた．p = 1.0 とし，temperature は 1.0 として推論した．

GECToR　GECToR は，[20] に従って 3 ステージで学習した．データセットは，1 段階目は PIE-synthetic．2 段階目は BEA-2019 の学習データから少なくとも一つの訂正が行われるペアのみを用いたもの，3 段階目は W&I-LOCNESS の学習データ全てを用いた．モデルには `bert-base-cased` と `deverta-v3-large` を用いた．学習の設定も基本的に [20] に従うが，1 段階目は 10epochs とした．推論時は，keep confidence と minimum error probability threshold をデータセットごとに決定する．両方の値を 0.1 から 1.0 までを試行し，開発データの ERRANT F0.5 が最大となる値を用いて評価データの推論に用いる．我々の再現結果は，Omelianchuk ら [20] や Tarnavskyi ら [6] の値と競合的な結果である．

T5　T5 は [22] に従って，cLang データセット 2,372,119 文を用いて追加学習した．ベースモデルには `t5-v1_1-large` を用いた．学習率を 1e-5 とし，10 エポック学習した．我々の再現の結果は Rothe ら [22] の結果を超えた．例えば，Table 1 に示した BEA-2019 test set のスコア $F_{0.5}$ = 73.4 は，Rothe ら [22] が報告する T5-large のスコア $F_{0.5}$ = 72.06 を超えている．

B　テンプレート詳細

4.3 節で使用した 10 種類のテンプレートを表 4 に示す．

表 4　本稿の実験で使用した 10 種類のテンプレート．[SOURCE] は誤り文に，[FEWSHOT] は few-shot 事例に置き換わる．

ID	テンプレート
1	You are a grammatical error correction tool. Your task is to correct the grammaticality and spelling in the input sentence. Make the smallest possible change in order to make the sentence grammatically correct. Change as few words as possible. Do not rephrase parts of the sentence that are already grammatical. Do not change the meaning of the sentence by adding or removing information. If the sentence is already grammatically correct, you should output the original sentence without changing anything. Return only the corrected text and nothing more.\n[FEWSHOT]\nInput sentence: [SOURCE]\nOutput sentence:
2	Make minimal changes to the following text such that it is grammatically correct. Return only the corrected text and nothing more.\n[FEWSHOT]\nInput sentence: [SOURCE]\nOutput sentence:
3	Please correct the following text. Do not attempt to rewrite it into perfect English or to interpret the text. Often, things could be expressed better by paraphrase, but the task is to make minimal changes to correct the text. Do not change anything that is correct. Please make no changes if there are no errors. Return only the corrected text and nothing more.\n[FEWSHOT]\nInput sentence: [SOURCE]\nOutput sentence:
4	Reply with a corrected version of the input sentence with all grammatical and spelling errors fixed. If there are no errors, reply with a copy of the original sentence. Return only the corrected text and nothing more.\n[FEWSHOT]\nInput sentence: [SOURCE]\nOutput sentence:
5	Correct the grammatical errors in the following sentence. Return only the corrected text and nothing more.\n[FEWSHOT]\n[SOURCE]; output:
6	Revise mistakes in this text. Return only the corrected text and nothing more.\n[FEWSHOT]\n[SOURCE]; output:
7	Rewrite the following text with proper grammar. Return only the corrected text and nothing more.\n[FEWSHOT]\n[SOURCE]; output:
8	Improve the grammar of this text. Return only the corrected text and nothing more.\n[FEWSHOT]\nInput sentence: [SOURCE]\nOutput sentence:
9	Correct the following to standard English. Return only the corrected text and nothing more.\n[FEWSHOT]\nSentence: [SOURCE]\nCorrection:
10	Fix the errors in this sentence. Return only the corrected text and nothing more.\n[FEWSHOT]\nInput sentence: [SOURCE]\nOutput sentence: